

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION
EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété
Intellectuelle
Bureau international



(43) Date de la publication internationale
21 juillet 2005 (21.07.2005)

PCT

(10) Numéro de publication internationale
WO 2005/066769 A2

(51) Classification internationale des brevets⁷ : **G06F 9/40**

(21) Numéro de la demande internationale :
PCT/EP2004/053292

(22) Date de dépôt international :
6 décembre 2004 (06.12.2004)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
03 14397 9 décembre 2003 (09.12.2003) FR

(71) Déposant (pour tous les États désignés sauf US)
: **THALES** [FR/FR]; 45, rue de Villiers, F-92200
NEUILLY-SUR-SEINE (FR).

(72) Inventeurs; et

(75) Inventeurs/Déposants (pour US seulement) :
BAILLEUL, Arnaud [FR/FR]; THALES Intellectual Property, 31-33, avenue Aristide Briand, F-94117 ARCUEIL Cedex (FR). **LE-SAUX, Thierry** [FR/FR]; THALES Intellectual Property, 31-33, avenue Aristide Briand, F-94117 ARCUEIL Cedex (FR).

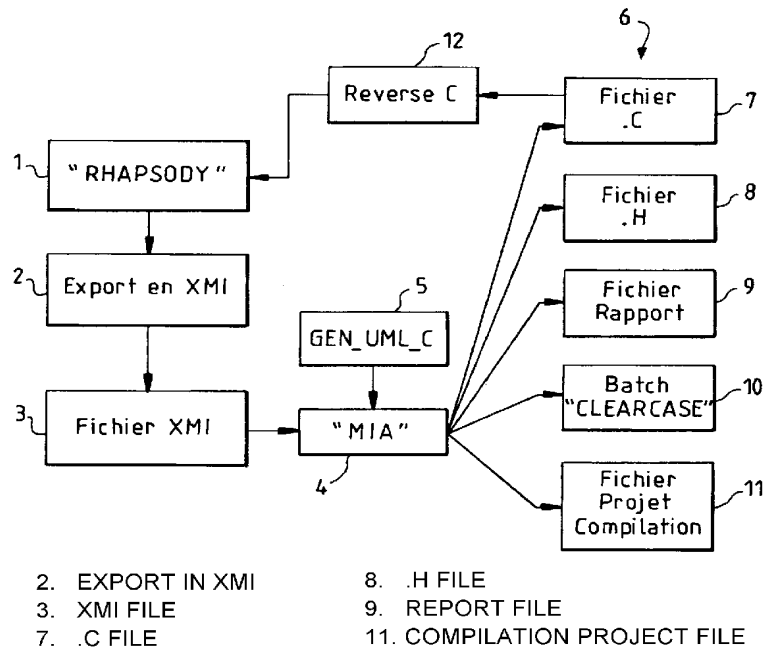
(74) Mandataires : **CHAVERNEFF, Vladimir** etc.; THALES Intellectual Property, 31-33, avenue Aristide Briand, F-94117 ARCUEIL Cedex (FR).

(81) États désignés (sauf indication contraire, pour tout titre de protection nationale disponible) : AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH,

[Suite sur la page suivante]

(54) Title: METHOD FOR GENERATING CODE C FROM UML SPECIFICATIONS

(54) Titre : PROCEDE DE GENERATION DE CODE C A PARTIR DE SPECIFICATIONS UML



(57) Abstract: The inventive method is characterised in that a detailed implementation model is produced in UML language, the data of said model is structured so that it can be used for the ModelInAction script generation tool (said scripts so-called MIA and produced by the company Sodifrance), and said tool is used to produce files in language C, namely files in .C, files in .H, a generation report file, batch files for configuration management, and compilation project files.

[Suite sur la page suivante]

WO 2005/066769 A2



PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

- (84) **États désignés** (sauf indication contraire, pour tout titre de protection régionale disponible) : ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), eurasién (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Publiée :

— sans rapport de recherche internationale, sera republiée dès réception de ce rapport

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) **Abrégé :** Le procédé conforme à l'invention est caractérisé en ce que l'on produit un modèle d'implémentation détaillé en langage UML, que l'on structure les données de ce modèle pour les rendre exploitables par l'outil de génération de scripts « ModelInAction » (dit « MIA » et produit par la société Sodifrance), et que l'on fait produire à cet outil des fichiers en langage C, à savoir des fichiers en .C, des fichiers en .H, un fichier de rapport de génération, des fichiers « batch » de gestion de configuration et des fichiers de projet de compilation.

PROCEDE DE GENERATION DE CODE C A PARTIR DE SPECIFICATIONS UML

La présente invention a pour objet un procédé de génération de code C à partir de spécifications UML.

Il existe des générateurs de code C à partir de code UML, tels que celui de la société I-LOGIX, mais le langage C, n'étant pas un code objet, on ne peut pas transcrire directement et automatiquement en code C des concepts « objet » tels que l'héritage et le polymorphisme. Cette impossibilité limite l'intérêt de l'utilisation de la modélisation en UML pour produire du code C. Les générateurs connus de code C ne produisent que des « squelettes » de code ou du code reflétant un usage très limité des concepts UML.

La présente invention a pour objet un procédé de génération de code C à partir de spécifications en langage UML d'un modèle, permettant de produire automatiquement la totalité du code C, aussi bien un code statique (y compris la génération de classes et de relations) qu'un code dynamique (en particulier pour les machines à états), le code produit respectant avantageusement les spécifications Do-178B niveau A. La présente invention a également pour objet un procédé qui garantisse la cohérence complète entre le code C généré et les spécifications écrites dans le modèle UML , qui permette de produire un rapport de génération de code C simultanément avec la génération de ce code, ainsi que des scripts de mise en configuration, avantageusement pour l'outil « Clearcase » de la société Rational-IBM. Dans le cas de systèmes embarqués, le code C ainsi produit pour de tels systèmes sera directement un logiciel embarqué.

Le procédé conforme à l'invention est caractérisé en ce que l'on produit un modèle d'implémentation détaillé en code UML, que l'on structure les données de ce modèle pour les rendre exploitables par l'outil de génération de scripts « Model In Action » (dit « MIA » et produit par la société Sodifrance), et que l'on fait produire à cet outil des fichiers en langage C, à savoir des fichiers « .C », des fichiers « .H », un fichier de rapport de génération, des fichiers « batch » de gestion de configuration et des fichiers de projet de compilation.

Selon une caractéristique avantageuse de l'invention, le code C généré recouvre 100% de la spécification UML du logiciel (le logiciel généré est le logiciel embarqué) à savoir que tout le spectre de génération est traité en statique (relations/classes) comme en dynamique (machines à états).

5 La présente invention sera mieux comprise à la lecture de la description détaillée d'un mode de mise en oeuvre, pris à titre d'exemple non limitatif et illustré par le dessin annexé, sur lequel :

-la figure 1 est un diagramme illustrant les étapes principales du procédé de l'invention, et

10 -la figure 2 est une vue partielle d'un exemple de rapport de génération, correspondant à un fichier de rapport de génération pouvant être produit selon le procédé de l'invention .

Dans l'exemple décrit ci-dessous, on établit, de façon classique, un modèle en langage UML à l'aide d'un outil (1) couramment utilisé à cet effet, par exemple l'outil « RHAPSODY » de la société I-LOGIX. Le modèle ainsi
15 créé est exporté (2) sous forme de fichier (3) en langage XML. La production du fichier XML se fait, par exemple, à l'aide d'un outillage d'export tel que le « XML Toolkit » de la société I-LOGIX. Le fichier 3 permet d'injecter la structure de données UML dudit modèle dans un moteur de génération de
20 fichiers (4). Ce moteur (4) est ici l'outil « Model In Action » (plus simplement dénommé « MIA ») de la société SODIFRANCE. Il est associé à une application de paramétrage de scripts (5), dénommée GEN_UML_C, réalisée par le Demandeur. Le moteur (4) met en oeuvre l'application (5) pour produire une série de cinq sortes de fichiers, référencée (6) dans son
25 ensemble, représentant le code C correspondant au modèle de départ.

La série 6 de fichiers comprend : des fichiers « .C » (7), des fichiers « .H » (8), un fichier (9) de rapport de génération de code (comme exigé par les spécifications Do précitées), un fichier « batch » (10) conforme aux
30 fichiers 7 , 8 et 11 étant produits de façon classique, ne seront pas décrits plus en détail.

Il est possible de modifier manuellement le corps des méthodes des fichiers « .C » générés. Lors des générations suivantes, ces modifications ne seront pas écrasées mais conservées et intégrées dans les nouveaux
35 fichiers générés. Dans une optique MDE (Model Driven Engineering) où le

modèle et non les fichiers sources sont le coeur du développement, il est préférable de remonter ces modifications manuelles dans le modèle. Il est alors possible d'intégrer ces modifications automatiquement dans le modèle à l'aide de l'outil de « roundtrip » appelé « ReverseC » (12).

5 Le fichier de rapport (9) permet de garder l'historique de la génération du code C et de comparer entre elles deux versions de rapports de génération. Dans le cas présent, le rapport est un fichier XML comportant toutes les informations correspondant au modèle UML, les fichiers et « packages » produits, les scénarios de génération utilisés,.... Un exemple
10 d'une vue d'écran d'un extrait d'un tel rapport (« Generation report ») est représenté en figure 2. A chaque élément de ce rapport, on associe un ou plusieurs totaux de contrôle (« checksums », par exemple des CRC sur 32 bits) permettant d'effectuer la comparaison entre versions différentes et la
15 nouvelle version de génération et une version de référence de génération.

A partir des comparaisons ainsi effectuées, on déduit différents états qui fournissent un « état de comparaison » du fichier examiné. Le tableau ci-dessous fournit ces états de comparaison. Dans ce tableau, les états figurent dans leur version anglaise, c'est-à-dire dans la version dans laquelle ils
20 apparaissent dans le rapport de génération.

Etats de comparaison	Description de l'état de comparaison du fichier
<i>New</i>	L'état « new » est réservé aux fichiers nouveaux par rapport à la version de référence.
<i>Unmodified</i>	S'applique lorsqu'aucune modification n'a été apportée à un fichier par rapport au fichier de référence.
<i>Modified</i>	S'applique lorsqu'il y a eu des modifications dues uniquement à des modifications du modèle UML par rapport au fichier de référence.
<i>Manually modified</i>	S'applique en cas de modifications manuelles par rapport à la version de référence.
<i>Modified & manually modified</i>	S'applique en cas de modifications des deux types d'un fichier par rapport à la version de référence.
<i>Removed</i>	S'applique lorsqu'un fichier n'existe plus dans une nouvelle version. Un fichier éliminé n'est plus pris en considération dans les rapports de génération ultérieurs.

Comme illustré en figure 2, un rapport de génération conforme à l'invention ("Generation Report") comporte dans des fenêtres en en-tête les deux informations suivantes:

- Etiquette du numéro de version du rapport de référence ("Reference report label"),
- Etiquette du numéro de version du rapport courant ("New report label"), provenant d'une version précédente de génération,

Ensuite, le corps du rapport comporte les trois colonnes suivantes: "Item" (désignation des différents éléments constitutifs et chemin d'accès dans leur unité de stockage), "Comparison status" (état de comparaison, conformément au tableau ci-dessus) et "Label" (étiquette, indiquant les numéros de version dans leur catégorie). Les éléments figurant dans la première colonne sont les suivants:

- Désignation du modèle UML ("Model"). Ici, son état est "modifié" et son numéro de version est 2.0,

5

- Désignation des ensembles logiciels ("Packages"). Dans l'exemple décrit ici, ce sont des ensembles Java et Clearcase. Ici, ils sont non modifiés et leur version est 1.0,
- 5 - Désignation des scénarios de génération ("Scenarios"). Dans l'exemple, ce sont des scénarios de génération Java et de "batch" Clearcase. Ici, ils sont non modifiés et leur version est 1.0,
- Désignation des fichiers générés ("Generated files"). Dans l'exemple, ces fichiers sont au nombre de sept, tous de version 2.0, le premier est nouveau, les statuts des autres étant divers,
10 comme indiqué à la deuxième colonne.

Le bas du rapport comporte une fenêtre indiquant le nom du scénario en cours ("batch Clearcase" dans le cas présent), un bouton "Generate" pour lancer la génération du scénario, et une première fenêtre "Files" indiquant les noms des fichiers de texte Clearcase générés dans ce
15 scénario, et une deuxième fenêtre "Generated text" affichant le texte de ces fichiers.

Les fichiers "batch" de Clearcase sont générés de façon à mettre à jour automatiquement le tableau Clearcase comportant tous les fichiers de code généré, conformément aux informations figurant dans le rapport de
20 génération. La mise à jour du tableau Clearcase est faite en deux phases, avec le lancement des deux fichiers PERL suivants:

- "checkout.txt": ce fichier prépare la mise à jour du tableau Clearcase en vérifiant tous les fichiers modifiés,
- "checkin.txt" : ce fichier représente le tableau Clearcase mis à jour.

REVENDICATIONS

1. Procédé de génération de code C à partir de spécifications UML, caractérisé en ce que l'on produit un modèle d'implémentation détaillé en code UML, que le modèle ainsi créé est exporté sous forme de fichier en langage XML, que ce
5 fichier XML est envoyé à un moteur de génération de fichiers qui est l'outil « Model In Action », que l'on associe cet outil à une application de paramétrage de scripts, et que l'on fait produire à cet outil des fichiers en langage C, à savoir des fichiers « .C », des fichiers « .H », un fichier de rapport de
10 génération, des fichiers « batch » de gestion de configuration et des fichiers de projet de compilation.
2. Procédé selon la revendication 1, caractérisé en ce que le code C généré recouvre 100% de la spécification UML du logiciel, tout le spectre de génération étant traité en statique
15 comme en dynamique
3. Procédé selon la revendication 1 ou 2, caractérisé en ce que le modèle est produit à l'aide d'un outil de modélisation UML
4. Procédé selon la revendication 3, caractérisé en ce que l'outil de modélisation UML est « RHAPSODY » de la société I-
20 LOGIX
5. Procédé selon l'une des revendications précédentes, caractérisé en ce que le fichier de rapport de génération comporte les informations suivantes :
 - numéro de version du rapport de référence,
 - 25 -numéro de version du rapport courant,
 - désignation du modèle UML avec son état et son numéro de version,
 - désignation des ensembles logiciels produits avec leur état et leur numéro de version,
 - 30 -désignation des scénarios de génération avec leur état et leur numéro de version,
 - désignation des fichiers générés avec leur état et leur numéro de version

7

-nom du scénario en cours,

-nom des fichiers de texte générés du scénario.

- 5 6. Procédé selon la revendication 5, caractérisé en ce que les scénarios sont du type « Clearcase ».
7. Procédé selon la revendication 5 ou 6, caractérisé en ce que les états des fichiers générés sont des états de comparaison par rapport à ceux d'une génération précédente (génération référence).
- 10 8. Procédé selon la revendication 6, caractérisé en ce que l'état de chaque fichier est l'un des suivants :
- nouveau,
- non modifié
- modifié
- modifié manuellement,
- 15 -modifié et modifié manuellement,
- éliminé.

20

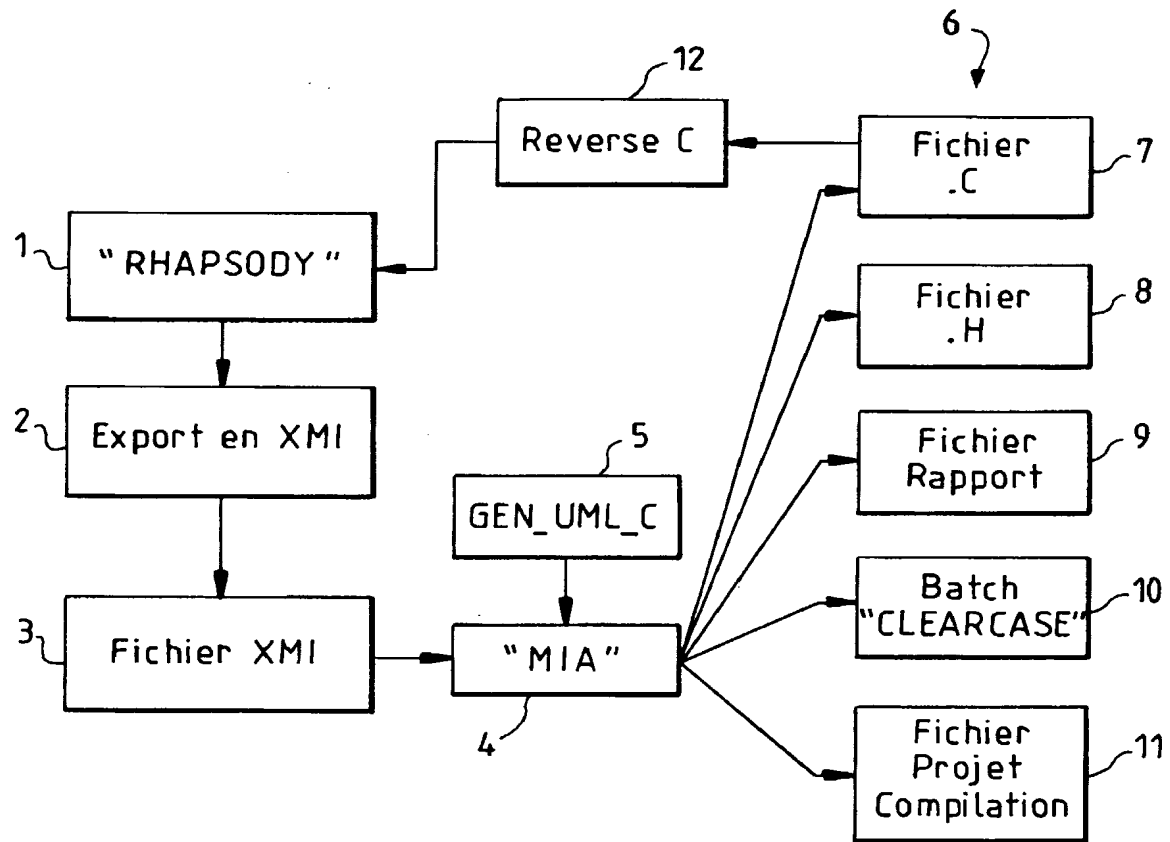


FIG.1

Generation report

File

Reference Report Label : 1.0

New Report Label : 2.0

Item	Comparison Status	Label
Generation report		
Model		
<ul style="list-style-type: none"> C:\Scriptor\UML\Examples\Models\Rose\Enterprise.mdl 	Modified	MODEL_2.0
Packages		
<ul style="list-style-type: none"> C:\Scriptor\UML\Examples\Scripts\Java.pkg C:\Scriptor\UML\Examples\Scripts\ClearCase.pkg 	Unmodified Unmodified	FWK_GEN_1.0 FWK_GEN_1.0
Scenarios		
<ul style="list-style-type: none"> C:\Scriptor\UML\Examples\Scripts\Java_generation.scn C:\Scriptor\UML\Examples\Scripts\Batch_ClearCase.scn 	Unmodified Unmodified	FWK_GEN_1.0 FWK_GEN_1.0
Generated Files		
<ul style="list-style-type: none"> c:\Generation\Scriptor\source\Batiment.java c:\Generation\Scriptor\source\Adresse.java c:\Generation\Scriptor\source\Bureau.java c:\Generation\Scriptor\source\Employe.java c:\Generation\Scriptor\source\Pdg.java c:\Generation\Scriptor\source\Service.java c:\Generation\Scriptor\source\Entreprise.java 	New Unmodified Unmodified Modified and Manually modified Modified Removed Modified	2.0 2.0 2.0 2.0 2.0 2.0 2.0

Batch ClearCase

Generate

Files

Generated text

Quit

FIG.2